AGU100 ADVANCING EARTH AND SPACE SCIENCE

JGR

Journal of Geophysical Research: Space Physics

TECHNICAL REPORTS: METHODS

10.1029/2018JA025274

Key Points:

- We use a deep neural network to automatically extract features from auroral images
- With these features we train a machine to predict the detailed auroral image category
- We achieve an auroral classification accuracy of 82% and an auroral detection rate of 96%

Correspondence to:
L. B. N. Clausen,
lasse.clausen@fys.uio.no

Citation:
Clausen, L. B. N., & Nickisch, H. (2018). Automatic classification of auroral images from the Oslo Auroral THEMIS (OATH) data set using machine learning. *Journal of Geophysical Research: Space Physics*, *123*, 5640–5647. https://doi.org/10.1029/2018JA025274

Received 26 JAN 2018
Accepted 1 JUN 2018
Accepted article online 9 JUN 2018
Published online 11 JUL 2018

Automatic Classification of Auroral Images From the Oslo Auroral THEMIS (OATH) Data Set Using Machine Learning

Lasse B. N. Clausen[1] and Hannes Nickisch[2]

[1] Department of Physics, University of Oslo, Oslo, Norway, [2] Philips Research, Hamburg, Germany

Abstract Based on their salient features we manually label 5,824 images from various Time History of Events and Macroscale Interactions during Substorms (THEMIS) all-sky imagers; the labels we use are *clear/no aurora*, *cloudy*, *moon*, *arc*, *diffuse*, and *discrete*. We then use a pretrained deep neural network to automatically extract a 1,001-dimensional feature vector from these images. Together, the labels and feature vectors are used to train a ridge classifier that is then able to correctly predict the category of unseen auroral images based on extracted features with 82% accuracy. If we only distinguish between a binary classification *aurora* and *no aurora*, the true positive rate increases to 96%. While this study paves the way for easy automatic classification of all auroral images from the THEMIS all-sky imager chain, we believe that the methodology shown here is readily applied to all images from any other auroral imager as long as the data are available in digital form. Both the neural network and the ridge classifier are free, off-the-shelf computer codes; the simplicity of our approach is demonstrated by the fact that our entire analysis comprises about 50 lines of Python code. Automatically attaching labels to all available all-sky imager data would enable statistical studies of unprecedented scope.

1. Introduction

Aurora Borealis and Aurora Australis are arguably the most impressive manifestations of solar wind/magnetosphere coupling. They are caused by charged particles (mostly electrons but also protons) originating from near-Earth space that have been accelerated along magnetic field lines toward Earth and subsequently collide with neutral constituents (mostly atomic oxygen) of the upper atmosphere. Since the vast space that is the magnetosphere maps along magnetic field lines into the upper atmosphere, it acts as a screen onto which magnetospheric dynamics are projected. Hence, observing the aurora from the ground allows one to study large-scale magnetospheric processes both on the day but also on the nightside.

Ground-based auroral data were instrumental in establishing the substorm concept (Akasofu, 1964), one of the major modes by which the magnetosphere dissipates energy (Akasofu, 1981; Clausen et al., 2014). As defined by Akasofu (1964), a substorm consists of two phases: the break-up phase and the recovery phase. During breakup a single dim arc suddenly brightens and large regions of the nightside sky abruptly fill with bright, discrete aurora, lasting for about 10 min or so. During the recovery phase, the break-up aurora dims, becomes patchier and diffuse, and eventually completely fades. Later it was established (Bargatze et al., 1985; McPherron et al., 1973) that a third phase precedes the break-up phase. Since during this first phase energy is loaded into the magnetospheric tail through dayside magnetic reconnection, this phase has been termed the growth phase. The arc that eventually becomes the break-up arc typically moves equatorward during this time.

Since the aurora is formed through processes in near-Earth space, it is clear, then, that the morphology of auroral forms as observed from the ground is integral to our understanding of magnetospheric dynamics. It would therefore be desirable to automatically classify the vast amount of existing ground-based auroral data in order to enable large statistical studies.

Automatic auroral image classification has already used a number of techniques from computer vision, pattern recognition, and machine vision with a strong emphasis on hand-designed features. First attempts used a two-step classification based on sparse edges and skeletons (Syrjäsuo et al., 2001) for individual images. Following up on these early approaches, *k*-nearest neighbor classification and principal component analysis of shapes were used for auroral tracking (Syrjäsuo & Donovan, 2002), which later allowed to automatically assess auroral occurrence statistics (Syrjäsuo & Donovan, 2004). Finally, auroral features were further refined to Fourier descriptors based on explicit shape models using contour and edge detection (Syrjäsuo et al., 2007). Once trained, the automatic classification strategies used in these studies were typically able to correctly distinguish between images showing aurora and images not showing aurora in 85% to 96% of the cases.

Another line of work used features from local binary patterns and scale-invariant feature transforms which were classified by a (nonlinear) support vector machine (Rao et al., 2014). Using color images from all-sky imagers across Finland, they were able to achieve true-positive rates of about 90% when using the labels *no aurora*, *aurora*, and *cloudy*.

Yang et al. (2012) used a hidden Markov model to classify data from an imager located on Svalbard. They explicitly included information about temporal dynamics using sequences of auroral images. They found that by this inclusion, they can achieve a detection rate of up to 85% depending on the length of the time series. Their categories were *arc*, *drapery*, *hot spot*, and *radial*; due to the imager's location, the last category is somewhat specific to auroral images taken at *cusp* latitudes.

Over the last few years, the fields of computer vision and machine learning have seen a big methodological paradigm shift: The focus from small-scale data sets and algorithms relying on hand-crafted features has moved to large-scale data sets and learning machines that automatically extract the feature representation from the raw data. Fueled by the widespread availability of curated data sets, associated benchmarks, fully trained models, and well-designed software libraries, as well as the growing computational power of modern graphics cards, there has been a tremendous wave of excitement and success in these communities. As a result, the entrance barrier for using deep neural network models in practice has become very low. In a neural network, artificial neurons are arranged in layers between the input (in our case an auroral image) and the output (in our case an image classification). The term *deep* loosely refers to networks that benefit from two improvements over earlier, *normal* networks. First, due to increasing computing power, deep neural networks can have significantly more layers than earlier; note, however, that there exists no fixed threshold in the number of layers between a normal and a deep neural network. Furthermore, deep refers to networks that have been trained using recent algorithms, that is, that benefit from recent research regarding the algorithmic implementation of neural networks. While the previous studies mentioned above use extensive amounts of hand-crafted computer code, our analysis uses freely available, standard machine learning libraries, and off-the-shelf tools. The results presented here are indeed produced by about 50 lines of Python code.

2. Methodology

Our goal is to train a machine such that it can automatically classify auroral images depending on the observed features. Due to its widespread use and convenient availability we choose data from the Time History of Events and Macroscale Interactions during Substorms (THEMIS) all-sky imagers (Donovan et al., 2006); it should be noted, though, that we believe that the results from this study are easily transferred to other auroral data sets.

The algorithms we use to classify auroral images all fall into the category of supervised learning; in supervised learning the goal is to train a model from labeled training data. Once the model is optimized, it can be used to make predictions about unseen or future data. The first step, then, is to create a training data set of labeled auroral images.

2.1. Labels

Based on our experience, we choose to introduce $L = 6$ labels $y \in \{0, 1, 2, 3, 4, 5\}$ which cover the range of phenomena observed in ground-based auroral imaging.

Although there is great overlap between our categories and those used in earlier studies, they are not exactly congruent. For example, Syrjäsuo and Donovan (2004) used four categories to describe the auroral displays observed from the all-sky imager at Gillam: no aurora, arcs, patchy aurora, and omega-bands. Their *patchy* is very similar to our diffuse, and the last category is the only one that differs from ours, although most of the images identified as *omega-band* would in our scheme probably be classified as discrete.

2.2. Image Preparation

As mentioned above, the images in our training data set originate from the THEMIS all-sky imager network. Using the quick-look plots available on the THEMIS website, we select by hand 84 intervals comprising 126 hr which represent all six of the phenomena listed above. In total, 5,824 images are selected at random from

y	Label	Explanation
0	*arc*	This label is used for images that show one or multiple bands of aurora that stretch across the field-of-view; typically, the arcs have well-defined, sharp edges.
1	*diffuse*	Images that show large patches of aurora, typically with fuzzy edges, are placed in this category. The auroral brightness is of the order of that of stars.
2	*discrete*	The images show auroral forms with well-defined, sharp edges, that are, however, not arc like. The auroral brightness is high compared to that of stars.
3	*cloudy*	The sky in these images is dominated by clouds or the dome of the imager is covered with snow.
4	*moon*	The image is dominated by light from the Moon.
5	*clear/noaurora*	This label is attached to images which show a clear sky (stars and planets are clearly visible) without the appearance of aurora.

these intervals and then processed. In the first step of the processing the raw auroral image is cropped in size by 15% in order to remove pixels that correspond to very low elevation angles. In order to enhance dim features, the brightness of each image is scaled to a value in the interval [0, 1]. This is done by first calculating the 1st percentile brightness value for each image individually and then subtracting this value from each pixel within the image. Then, the 99th percentile brightness value is computed (from the now altered brightness values of each image), and each brightness value is divided by it. Finally, all values below 0 are set to 0 and all values above 1 are set to 1. The cropped, scaled version of each image is used to decide which of the six labels y to attach to it.

In Figure 1 we show some examples from our training set. In panels (a)–(c) we show representative examples from the cloudy, moon, and clear/noaurora category. Examples from the arc, discrete, and diffuse categories are shown in the second row (panels (c)–(e)).

It should be clear to anyone who has worked with ground-based auroral data that the categories introduced above are by no way mutually exclusive, as the top two rows of Figure 1 might suggest. A few examples of problematic images are shown in the bottom row where it is unclear whether the displayed auroral phenomenon is arc or diffuse (panel g), diffuse or discrete (panel h), cloudy or moon (panel i). Ambiguity notwithstanding, we assign the labels in our training set to the best of our abilities. It should also be noted that to the trained human eye, there is a clear distinction between an image in any of the three auroral categories (arc, discrete, and diffuse) and an image in any of the other three (cloudy, moon, and clear/noaurora).

In principle, we could now train a deep neural network with these labeled images. Training a deep neural network from scratch, however, is very expensive both in terms of computational complexity and required number of data points. We therefore, as a first step, choose to use a pretrained deep neural network to automatically extract features from the cropped and scaled image. Furthermore, before feature extraction we also rotate each image by a random angle around its center in order to not bias the feature selection toward a certain orientation of structures like star constellations or east-west aligned arcs. These extracted features are represented by a 1,001-element feature vector $\mathbf{f}$.

2.3. Feature Extraction

We compute the feature vector $\mathbf{f}$ from each image x using TensorFlow^{TM} (Abadi et al., 2015), an open-source software library for numerical computation originally developed by researchers and engineers from the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research. We use the latest Inception-v4 (Szegedy et al., 2017) pretrained neural network (dated 9 September 2016), which offers the best compromise between classification accuracy and computational complexity to date. The neural network has been trained on the ILSVRC-2012-CLS image classification data set (Russakovsky et al., 2015) which contains about 1.2 million labeled images of objects from 1,000 different categories like pandas, container ships, and dandelions.

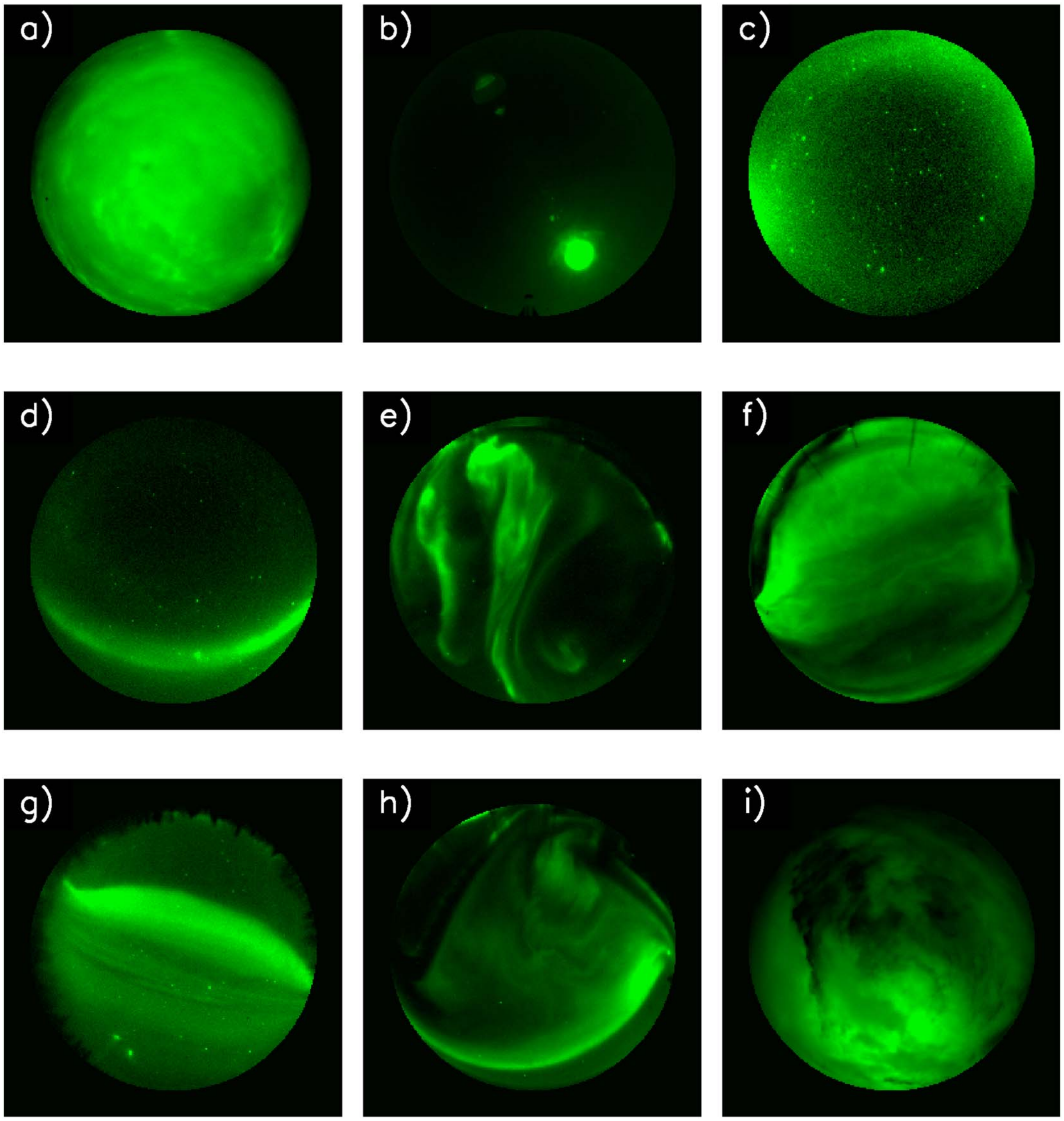

Figure 1. In panel (a) through (f) we show examples of processed auroral images for the categories *cloudy*, *moon*, *clear/noaurora*, *arc*, *discrete*, and *diffuse*. The bottom row shows examples where the category to assign to each image is ambiguous.

As described above, the idea behind using a pretrained neural network as feature extractor φ for the auroral images x_i with $i = 1 \dots 5824$ is to compute a feature representation $\varphi(x_i) =: \mathbf{f}_i \in \mathbb{R}^F$. Hence, all $N = 5824$ images need to be pushed through the pretrained network and the output is collected in a matrix of size 5824×1001. For later computations, the images x_i are no longer needed. Once $\mathbf{f}_i$ is computed, one of the possible discrete labels $y_i \in \{0, 1, 2, 3, 4, 5\}$ is assigned to each $\mathbf{f}_i$ to form the pairs $(\mathbf{f}_i, y_i)$. It is these pairs $(\mathbf{f}_i, y_i)$ that are then used to train another machine (distinctively different from the feature extraction neural network) to compute the mapping from the features $\mathbf{f}_i$ to the class labels y_i. This process is schematically shown in Figure 2.

2.4. Ridge Classification

There are a variety of supervised learning machines for classification. We use a method called ridge classification; although this method is standard within the machine learning community (e.g., Raschka, 2015a), we repeat its salient features here.

Ridge classification is a linear method extending and generalizing ordinary linear regression in two aspects: First, the added *ridge* improves the generalization capabilities of the method and second; it deals with binary labels rather than real-valued labels. Given a set of N F-dimensional input vectors $\mathbf{f}_i \in \mathbb{R}^F$ and N measurements

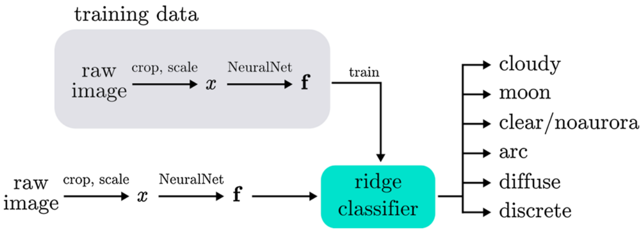

Figure 2. Schematic flow of the data, from the original raw image to the automatically attached label.

$y_i \in \mathbb{R}$, (ordinary) linear regression aims to find parameters $\mathbf{w} \in \mathbb{R}^F$—called weights—that minimize the least squares cost function $\mathcal{C}_{LS}$ measuring the agreement between labels y_i and linear prediction $\mathbf{w}^\top \mathbf{f}_i$

$$\mathcal{C}_{LS}(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \mathbf{w}^\top \mathbf{f}_i)^2 \tag{1}$$

where the superscript ⊤ denotes the matrix transpose. Note that a bias term $\mathbf{w}^\top \mathbf{f}_i + b$ can be included in the weight vector $\mathbf{w}$ by appending a constant entry to the feature vector $\mathbf{f}_i \leftarrow [\mathbf{f}_i; 1]$. When the number of features F is large and approaching the number of measurements N, ordinary linear regression risks to produce a model (i.e., a set of weights) that represents the training data set $(\mathbf{f}_i, y_i)$ well; however, it performs poorly when predicting labels of unseen data; in machine learning parlance this is called *overfitting*. In order to counterbalance overfitting in ordinary regression, several strategies are available. In ridge regression the strategy—called regularization—is to penalize large values of $\mathbf{w}$ measured by the squared L2 norm $\mathbf{w}^\top \mathbf{w}$ yielding the ridge regression cost function

$$\mathcal{C}_{RR}(\mathbf{w}) = \lambda \cdot \mathbf{w}^\top \mathbf{w} + \frac{1}{N} \sum_{i=1}^{N} (y_i - \mathbf{w}^\top \mathbf{f}_i)^2, \; \lambda \in \mathbb{R}_+. \tag{2}$$

Regularization in ridge regression thus limits the magnitude of the weights $\mathbf{w}$ (and hence the complexity of the learning machine). The scalar parameter λ balancing data fit and regularization is found through cross validation.

As the name suggests, ridge *regression* is used to estimate continuous real-valued y_i; however, ridge regression is easily turned into ridge (multiclass) classification where the model output is a set of L discrete values rather than a real number. This is done by combining L binary logistic regressors with weights $\mathbf{w}_c$ trained on binary labels $y_i^c \in \{0, 1\}$ in a one-against-the-rest fashion where $y_i^c = \delta_{y_i=c}$; this results in one set of weights $\mathbf{w}_c$ per class.

Using the nomenclature of our $N = 5824$ feature vectors $\mathbf{f}_i$ and auroral class labels y_i and combining the weights per class $\mathbf{w}_c$ into a matrix $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L]$, the ridge classification cost function takes the form

$$\mathcal{C}_{RC}(\mathbf{W}) = \lambda \cdot \mathrm{tr}(\mathbf{W}^\top \mathbf{W}) + \frac{1}{N} \sum_{i=1}^{N} H[\mathbf{e}_{y_i}, \mathrm{softmax}(\mathbf{W}^\top \mathbf{f}_i)] \tag{3}$$

where $H[\mathbf{p}, \mathbf{q}] = -\mathbf{p}^\top \log \mathbf{q} - (\mathbf{1} - \mathbf{p})^\top \log(\mathbf{1} - \mathbf{q}) = -\sum_{c=1}^{L} [p_c \log q_c + (1 - p_c) \log(1 - q_c)]$ is the binary cross-entropy function, $\mathbf{e}_k$ is a unit vector in dimension k and $\mathrm{softmax}(\mathbf{z}) = \frac{\exp(\mathbf{z})}{\mathbf{1}^\top \exp(\mathbf{z})}$ is the softmax function. The cost function is minimized using a second-order gradient descent algorithm (like Newton-Raphson) to find—in the above least squares sense—the optimal $\mathbf{W}$. Once the optimal weights $\mathbf{W}$ are found, unseen auroral images x can be classified by first extracting the feature vector $\mathbf{f}$ and then calculating the class probabilities

$$\mathbf{p} = \mathrm{softmax}(\mathbf{W}^\top \mathbf{f}) \in [0, 1]^L; \tag{4}$$

the class yielding the highest probability is then chosen as the predicted label.

Although multiclass classification seems disconnected from the (pretrained) neural network, it is possible (and common), to interpret the classifier as the *last layer* of a combined neural network. In other words, we have simply trained a neural network where only the parameters $\mathbf{W}$ of the last layer have been adjusted.

Table 1
Confusion Matrix of the Trained Ridge Classifier for One Particular Partion of the Training Set—Other Partitions Produce Very Similar Results

		Observed					
		Arc	Discrete	Diffuse	Cloudy	Moon	Clear/noaurora
Predicted	Arc	138	30	50	1	1	11
	Discrete	24	251	56	0	0	13
	Diffuse	26	31	353	1	0	2
	Cloudy	0	2	3	237	3	2
	Moon	0	0	2	3	188	3
	Clear/noaurora	15	10	13	0	0	278

It is worth emphasizing that TensorFlow$^{\text{TM}}$ including the feature extractor and the Inception-v4 checkpoint are freely available and can be run even on simple hardware. The ridge classification was done using the (also freely available) machine learning codes of the scikit-learn Python modules (Raschka, 2015b). Using these tools the code to extract features, train the ridge classifier, and classify unseen images comprises about 50 lines of code.

3. Results

In order to test the performance of any machine learning pipeline, one standard procedure (based on bootstrap resampling and similar to cross validation) is to partition the training set $(\mathbf{f}_i, y_i)$ randomly into two subsets: one is used for training as described above (typically 70% of the data points) and one subset (the remaining 30%) is used to compare the predictions of the trained machine with the actual labels. This process is repeated five times with different random partitions to obtain a measure of the model variance. Since the actual labels of all auroral images in the data set are known, we can compute what is called the confusion matrix (for one particular instance of the ridge classifier), shown in Table 1.

The values along the diagonal of the confusion matrix give the number of times the correct label was predicted. Off-diagonal values show where the model failed to predict the correct label. It can be seen from Table 1 that the model does particularly well in distinguishing between the nonauroral categories cloudy, moon, and clear/noaurora, that is, it rarely confuses an image of the Moon with one that shows clouds (or vice versa); it does less well in distinguishing between the different auroral labels. Intuitively, this is expected, as the feature overlap between the different auroral forms is larger.

Overall, we find that our model predicts 81.7 ± 0.1% of the cases correctly. When lumping the three auroral labels into one class and the nonauroral labels into a second class, our models correctly predicts the image label 95.60 ± 0.03% of the time.

3.1. Test Case

As a qualitative test of our model, we train the ridge classifier with the entire training data set and then classify 577 auroral images from Rankin Inlet selected at random between 0100 and 1000 UT on 21 January 2006. A keogram of the selected auroral images is shown in the lower panel of Figure 3. In the beginning of the interval, the sky is clear with a single arc forming within the field-of-view of the instrument which moves equatorward. At 0200 UT the arc disappears and the sky becomes clear again. Around 0333 UT an auroral activation occurs above the instruments, and the auroral activity continues until about 0800 UT. Soon after light cloud cover starts to drift into the field-of-view from the south.

Above the keogram in Figure 3 we show the probabilities the ridge classifier predicts for the six categories; the thicker the color at each time step, the larger the probability that the image belongs to this particular category.

When the arc appears just before 0200 UT, the probability of arc and diffuse increase significantly, while the probability of clear/noaurora decreases. This is in agreement with the first exemplary auroral image show in the top row. Once the arc disappears, the probability of clear/noaurora increases again, before decreasing again when the auroral activation is observed. Throughout the auroral activation at the center of this interval, the cumulative probability of the three auroral categories is around 80%.

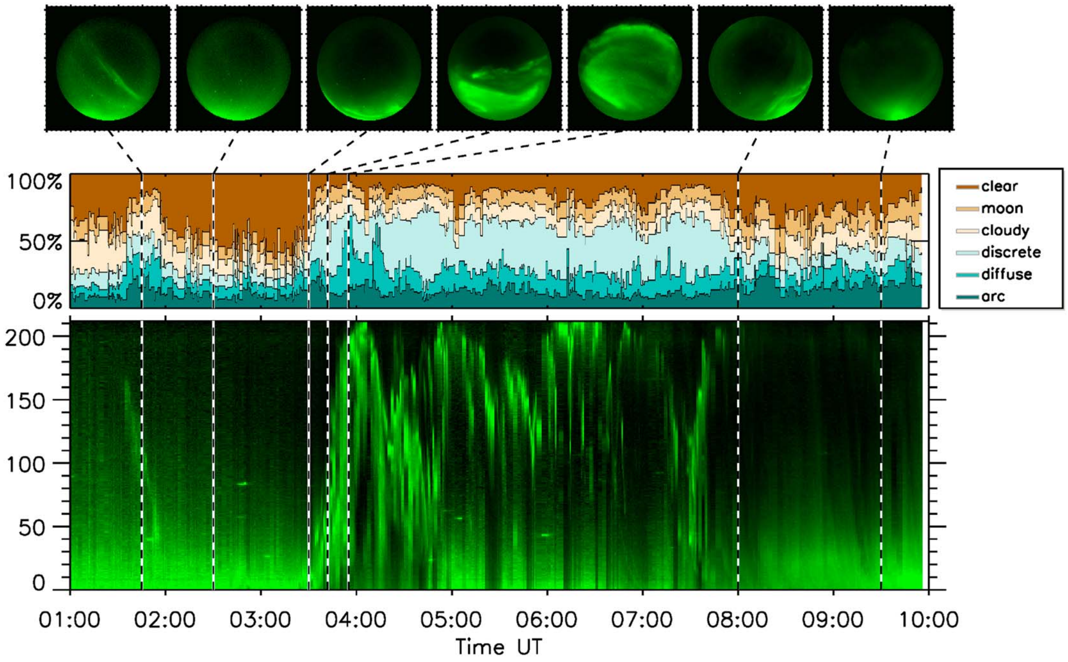

Figure 3. The bottom panel shows a keogram from auroral data collected on 21 January 2006 at Rankin Inlet. The middle panel shows the probabilities for the six categories as predicted by the ridge classifier trained with the entire training data set. At the top we show, for comparison, auroral images at different times.

The thin cloud cover toward the end of the interval is less well represented by the model; however, the cumulative probability of the auroral categories decreases to below 50%.

4. Summary and Outlook

From the THEMIS all-sky imager data set we have, based on the displayed features, manually labeled 5,824 auroral images from 84 representative intervals with one of the following categories: arc, discrete, diffuse, cloudy, moon, and clear/noaurora. We then used the feature detection mechanism of the pretrained TensorFlow^{TM} deep neural network to extract a 1,001-dimensional feature vector **f** from each image. These 5,824 feature vectors were then used to train a ridge classifier such that it is able to predict a label for unseen images. We find that the trained ridge classifier is able to predict 81.7 ± 0.1% of the image labels correctly. When only distinguishing between the presence and the absence of aurora (binary classification), our models correctly predict the image label 95.60 ± 0.03% of the time.

We believe that the methodology outlined in this study can easily be transferred to the entire THEMIS all-sky imager data set and also other auroral data sets. The advantage of our effort compared to earlier efforts in automatic classification of auroral images is, we believe, its reliance on battle-tested, ready-to-use software (TensorFlow^{TM}, Inception-v4, and the scikit-learn module) that is already in operational use in many commercial and scientific contexts. When applying the methodology outlined here to large data sets like that from the THEMIS all-sky imagers, the bottleneck will not be getting the software up and running. The bottleneck will be the speed of the feature vector extraction. For this study a standard laptop with an Intel i7 CPU was used for the feature extraction and classification. On this computer the feature extraction ran at about 1.5 images per second, such that the features of the entire training data set were extracted in about 1 hr. Here, however, significant speed-up can be expected through the use of powerful and optimized hardware. Training the ridge classifier on the same laptop is a matter of seconds.

The flexibility of the used software also allows a rapid implementation of different classifications. The labels used in this study are meant as suggestions, and clearly other labels could be used. Additional labels like *auroral activity with moon contamination*, *east-west aligned arc*, or *north-south aligned arc* could also be introduced. Even geolocation could be included in the labeling process in order to distinguish between poleward boundary arcs and polar cap arcs. All that would be needed is a training data set containing a few thousand images since both the automatic feature extraction and the ridge classifier training is independent of the used classifiers. We encourage the community to have a discussion about the appropriate classifications before a large undertaking such as automatically labeling all images within the THEMIS data set is started.

We did not yet exploit the fact that auroral images are captured as a time series rather than independent images in our experiments, which leaves room for further accuracy improvements, for example, by hidden Markov models or recurrent neural networks exploiting temporal correlations between subsequent images and labels. In particular, we expect improvements from temporal smoothness, that is, labels do typically not change at a rapid pace and labels tend to change in a particular order meaning that some label transitions are more likely than others a priori.

We believe that an automatic classification — using whatever set of labels appropriate — of large auroral data sets like the THEMIS all-sky imager data set would facilitate statistical studies of unprecedented scope, using literally tens of millions of images. Conceivably, this could change the way we use ground-based auroral images in magnetospheric research.

Acknowledgments
L. B. N. C. would like to acknowledge K. Rohleder and M. Baumgarten for making possible a fruitful collaboration. TensorFlow^{TM} is available at http://www.tensorflow.org. The feature extractor can be downloaded from https://github.com/tomrunia/TF_FeatureExtraction. The version-4 checkpoint of the pretrained Inception neural network is available under https://github.com/tensorflow/models/tree/master/research/slim. The scikit-learn Python module is available at https://scikit-learn.org. The THEMIS all-sky data are available through the THEMIS website at http://themis.ssl.berkeley.edu. The code and data allowing to reproduce our experiments along with detailed instructions are available from http://tid.uio.no/plasma/oath/.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems, software available from tensorflow.org.

Akasofu, S.-I. (1964). The development of the auroral substorm. *Planetary and Space Science*, *12*, 273–282. https://doi.org/10.1016/0032-0633(64)90151-5

Akasofu, S.-I. (1981). Energy coupling between the solar wind and the magnetosphere. *Space Science Reviews*, *28*, 121–190. https://doi.org/10.1007/BF00218810

Bargatze, L. F., Baker, D. N., Hones, E. W. Jr., & McPherron, R. L. (1985). Magnetospheric impulse response for many levels of geomagnetic activity. *Journal of Geophysical Research*, *90*, 6387–6394. https://doi.org/10.1029/JA090iA07p06387

Clausen, L. B. N., Milan, S. E., & Grocott, A. (2014). Thermospheric density perturbations in response to substorms. *Journal of Geophysical Research: Space Physics*, *119*, 4441–4455. https://doi.org/10.1002/2014JA019837

Donovan, E., Mende, S., Jackel, B., Frey, H., Syrjäsuo, M., Voronkov, I., et al. (2006). The THEMIS all-sky imaging array — System design and initial results from the prototype imager. *Journal of Atmospheric and Solar-Terrestrial Physics*, *68*, 1472–1487. https://doi.org/10.1016/j.jastp.2005.03.027

McPherron, R. L., Russell, C. T., & Aubry, M. P. (1973). Satellite studies of magnetospheric substorms on August 15, 1968. 9. Phenomenological model for substorms. *Journal of Geophysical Research*, *78*, 3131–3149. https://doi.org/10.1029/JA078i016p03131

Rao, J., Partamies, N., Amariutei, O., Syrjäsuo, M., & van de Sande, K. E. (2014). Automatic auroral detection in color all-sky camera images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *7*(12), 4717–4725.

Raschka, S. (2015a). *Python machine learning, chap. Predicting continuous target variables with regression analysis* (pp. 277–309). Birmingham: Packt Publishing Ltd.

Raschka, S. (2015b). *Python machine learning*. Birmingham: Packt Publishing Ltd.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, *115*(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

Syrjäsuo, M., & Donovan, E. (2002). Analysis of auroral images: Detection and tracking. *Geophysica*, *38*, 3–14.

Syrjäsuo, M., & Donovan, E. (2004). Diurnal auroral occurrence statistics obtained via machine vision. *Annales Geophysicae*, *22*, 1103–1113. https://doi.org/10.5194/angeo-22-1103-2004

Syrjäsuo, M., Donovan, E., Qin, X., & Yang, Y. (2007). Automatic classification of auroral images in substorm studies. In *International Conference on Substorms (ICS8)* (pp. 309–313). Alberta, Canada: University of Calgary.

Syrjäsuo, M. T., Kauristie, K., & Pulkkinen, T. I. (2001). A search engine for auroral forms. *Advances in Space Research*, *28*, 1611–1616. https://doi.org/10.1016/S0273-1177(01)00492-6

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, inception-ResNet and the impact of residual connections on learning, AAAI. pp. 4278–4284.

Yang, Q., Liang, J., Hu, Z., & Zhao, H. (2012). Auroral sequence representation and classification using hidden Markov models. *IEEE Transactions on Geoscience and Remote Sensing*, *50*, 5049–5060. https://doi.org/10.1109/TGRS.2012.2195667